

IN THE SPECIFICATION

Please amend the paragraph beginning at page 1, line 21 as follows:

Some electronic documents include text and annotation elements which indicate the semantics, hierarchy, structure, or format of the documents. The annotation elements, known as markups, within a document generally conform to a markup language which defines a set of annotation elements. The markup language defines which elements in the language are required elements, which elements are optional elements, and how annotation elements distinguish from neighboring text. Examples of markup languages include Standard Generalized Markup Language (SGML), Extensible Markup Language (XML), and hypertext markup language (HTML).

Please amend the paragraph beginning at page 2, line 4 as follows:

Conventionally, marking up documents according a markup language entails inputting the document into a specific custom-conversion program designed for marking up documents in the markup language. Examples of custom conversion programs are programs created using tools such as Omnimark and Balise. Thus, for example, marking up a document in SGML requires use of an SGML conversion program and marking up a document in HTML requires use of an HTML conversion program. In other words, the conventional approach to marking up documents ~~using~~ uses DTD-specific conversion programs.

Please amend the paragraph beginning at page 2, line 13 as follows:

This conventional approach suffers from at least five problems. First, because the converters are dependent on the structure of a single DTD, they cannot be used to markup documents according to other markup languages. Second, the converter cannot easily adapt to changes to its corresponding DTD, since the grammatical and semantic rules of the DTD are hard-coded into the converter, requiring the logic of the converter to be reprogrammed. Third, the hard-coded DTD semantics in the converter increases its size and complexity, and thus reduce its reliability. Fourth, the dependency of the converter on a specific DTD also reduces the

reusability of its source code for other DTDs. And fifth, conventional converters follow an all-or-nothing approach to markup, which prevents them from outputting a document with marked and unmarked portions. This restriction reduces the flexibility and application of the ~~translator~~ converter.